CRYPTOGRAPHIC ENTROPY TREE

TECHNICAL FIELD

[0001] The disclosure generally relates to generating and storing keys for securing data.

BACKGROUND

[0002] Many computing devices use cryptographic keys to secure data stored in device memory. Data can be encrypted using a cryptographic algorithm. Once encrypted, the data may only be readable by users in possession of the correct key. To safeguard against unauthorized data access, computing devices should keep the keys secure.

[0003] A computing device may employ several keys. For example, a computing device may have a plurality of user accounts. Each user account can require a separate key for access to the device's file system and/or applications. Specific portions of the device's file system may be secured by keys. Specific applications may require keys to run and/or to provide certain functions. In some implementations, not only may the file system and individual applications require separate keys, but these keys may differ among user accounts. For example, user 1 may use key 1 to access application 1, but user 2 may use key 2 to access application 1.

SUMMARY

[0004] Some computing devices may include a secure enclave processor (SEP) and an SEP memory. The SEP memory may be accessible to the SEP but inaccessible to other elements of the device, such as the main processor of the device. To prevent unauthorized users from obtaining keys to which they are not entitled, these computing devices may store keys in SEP memory. However, the SEP memory may only provide a limited amount of storage space for the keys.

[0005] Systems and methods described herein may extend the available storage space for keys by storing keys in generally accessible device storage. For example, the accessible keys may be stored in the device's file system. The accessible keys may be encrypted by the SEP, and the key or keys for decrypting the accessible keys may be stored in SEP memory. Accordingly, the SEP may protect the accessible keys without storing them in SEP memory.

[0006] Particular implementations provide at least the following advantages. The SEP can provide security by encrypting accessible keys using a combination of a general SEP key and entropy (e.g., a random number or random data value generated by the SEP). Keys can be isolated to separate operating system, service, and/or user instances because SEP may encrypt keys for each instance using different entropies. Accordingly, separate accounts can be separately encrypted, providing cryptographic isolation for files only accessible to specific users. Using entropy to generate keys can prevent one user from guessing another user's key, because the key is random rather than counter-based. Storing keys in accessible memory and securing the accessible keys with the SEP can extend the number of keys that can be used beyond the number that can be stored in SEP memory. Keys in accessible memory can be invalidated in bulk by resetting stored entropy values in SEP memory.

[0007] Details of one or more implementations are set forth in the accompanying drawings and the description below. Other features, aspects, and potential advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0008] FIG. **1** is a block diagram of an example device comprising a secure enclave processor (SEP).

[0009] FIGS. **2**A and **2**B are diagrams of example entropy trees.

[0010] FIG. **3**A illustrates an example of root key creation.

[0011] FIG. **3**B illustrates an example of initial key creation for entropy stored in SEP memory.

[0012] FIG. **3**C illustrates an example of initial entropy creation for entropies stored in a file system.

[0013] FIG. **3**D illustrates an example of entropy decryption.

[0014] FIG. **3**E illustrates an example of key creation and decryption upon device boot.

[0015] FIG. **3**F illustrates an example of key invalidation.

[0016] FIGS. **4**A-**4**D illustrate example fault tolerance features.

[0017] FIG. **5** is a flow diagram of an example key creation process.

[0018] FIG. **6** is a flow diagram of an example error process.

[0019] FIG. **7** is a flow diagram of an example post-boot process.

[0020] FIG. **8** is a flow diagram of an example anti-replay process.

[0021] FIG. **9** is a block diagram of an example system architecture implementing the features and processes of FIGS. **1**-**8**.

[0022] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

SEP-Equipped Devices

[0023] Computing devices described herein may include a plurality of data sets that can be independently secured. For example, a computing device may include a plurality of operating system (OS) partitions, with each partition requiring a different key to access. Each OS partition may include a plurality of device services and/or applications, each of which may require a different key to access. Each service and/or application may in turn include a plurality of user accounts, each of which may require a different key to access.

[0024] The computing device may use a secure enclave processor (SEP) to encrypt and decrypt each key. For example, the SEP may be configured to create keys and encrypt them using a root key and an entropy value (e.g., a random number). When a user wishes to access data secured by a key, the user may supply a credential to the SEP. If the credential is correct, the SEP can decrypt the key. The SEP can use secure, encrypted memory that is isolated from other computing device systems. Accordingly, the SEP can provide security for each partition, service, and/or user separately without allowing a user only authorized for one OS partition to gain access to another OS partition, for example. Systems and methods described herein allow the SEP to